

1Z0-809 Dumps

Java SE 8 Programmer II

<https://www.certleader.com/1Z0-809-dumps.html>



NEW QUESTION 1

Given:

```
class Book { int id;  
String name;  
public Book (int id, String name) { this.id = id;  
this.name = name;  
}  
public boolean equals (Object obj) { //line n1 boolean output = false;  
Book b = (Book) obj;  
if (this.name.equals(b.name)) output = true;  
}  
return output;  
}  
}
```

and the code fragment:

Book b1 = new Book (101, "Java Programing"); Book b2 = new Book (102, "Java Programing"); System.out.println (b1.equals(b2)); //line n2 Which statement is true?

- A. The program prints true.
- B. The program prints false.
- C. A compilation error occur
- D. To ensure successful compilation, replace line n1 with: boolean equals (Book obj) {
- E. A compilation error occur
- F. To ensure successful compilation, replace line n2 with: System.out.println (b1.equals((Object) b2));

Answer: A

NEW QUESTION 2

Given the code fragment:

```
5. IntConsumer consumer = e -> System.out.println(e);  
6. Integer value = 90;  
7. /* insert code fragment here */  
8. consumer.accept(result);
```

Which code fragment, when inserted at line 7, enables printing 100?

- A. Function<Integer> funRef = e -> e + 10; Integer result = funRef.apply(value);
- B. IntFunction funRef = e -> e + 10; Integer result = funRef.apply (10);
- C. ToIntFunction<Integer> funRef = e -> e + 10; int result = funRef.applyAsInt (value);
- D. ToIntFunction funRef = e -> e + 10; int result = funRef.apply (value);

Answer: A

NEW QUESTION 3

Which code fragment is required to load a JDBC 3.0 driver?

- A. Connection con = Connection.getDriver ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- B. Class.forName("org.xyzdata.jdbc.NetworkDriver");
- C. Connection con = DriverManager.getConnection ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- D. DriverManager.loadDriver ("org.xyzdata.jdbc.NetworkDriver");

Answer: B

NEW QUESTION 4

Given the code fragment:

```
for (Course a : Course.values()) {  
    System.out.print(a + " Fees " + a.getCost()+" " );  
}
```

Which is the valid definition of the Course enum?

```
A. enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}

B. enum Course { JAVA(100), J2ME(150);
    private static int cost;
    private Course(int c) {
        this.cost = c;
    }
    static int getCost() {
        return cost;
    }
}

C. final enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
    void setCost(int c) {
        this.cost = c;
    }
}

D. enum Course { JAVA(100), J2ME(150);
    private int cost;
    Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

NEW QUESTION 5

Which statement is true about java.time.Duration?

- A. It tracks time zones.
- B. It preserves daylight saving time.
- C. It defines time-based values.
- D. It defines date-based values.

Answer: C

NEW QUESTION 6

Given:

```
class Resource implements AutoCloseable {  
    public void close() throws Exception {  
        System.out.print("Close-");  
    }  
    public void open() {  
        System.out.print("Open-");  
    }  
}
```

and this code fragment:

```
Resource res1 = new Resource();  
try {  
    res1.open();  
    res1.close();  
} catch (Exception e) {  
    System.out.println("Exception - 1");  
}  
try (res1 = new Resource()) { // line n1  
    res1.open();  
} catch (Exception e) {  
    System.out.println("Exception - 2");  
}
```

What is the result?

- A. Open-Close- Exception - 1 Open-Close-
- B. Open-Close-Open-Close-
- C. A compilation error occurs at line n1.
- D. Open-Close-Open-

Answer: C

NEW QUESTION 7

Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1  
instantiation */  
ProductCode<Number, String> c2 = new ProductCode<Number, String>(); /* c2  
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must allow c1 instantiation to succeed and cause a compilation error on c2 instantiation.

Which definition of ProductCode meets the requirement?

```
A. class ProductCode<T, S<Integer>> {
    T c1;
    S c2;
}

B. class ProductCode<T, S extends T> {
    T c1;
    S c2;
}

C. class ProductCode<T, S> {
    T c1;
    S c2;
}

D. class ProductCode<T, S super T> {
    T c1;
    S c2;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

NEW QUESTION 8

Given the code fragment:

```
Path p1 = Paths.get("/Pics/MyPic.jpeg"); System.out.println (p1.getNameCount() + ":" + p1.getName(1) +
":" + p1.getFileName());
```

Assume that the Pics directory does NOT exist.

What is the result?

- A. An exception is thrown at run time.
- B. 2:MyPic.jpeg: MyPic.jpeg
- C. 1:Pics:/Pics/ MyPic.jpeg
- D. 2:Pics: MyPic.jpeg

Answer: B

NEW QUESTION 9

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom"); System.out.println (
// line n1
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. listVal.stream().filter(x -> x.length()>3).count()
- B. listVal.stream().map(x -> x.length()>3).count()
- C. listVal.stream().peek(x -> x.length()>3).count().get()
- D. listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()

Answer: A

NEW QUESTION 10

Given:

```
public class Customer { private String fName; private String lName; private static int count;
public customer (String first, String last) {fName = first, lName = last;
++count;}
static { count = 0; }
public static int getCount() {return count; }
}
```

```
public class App {
public static void main (String [] args) { Customer c1 = new Customer("Larry", "Smith");
Customer c2 = new Customer("Pedro", "Gonzales"); Customer c3 = new Customer("Penny", "Jones"); Customer c4 = new Customer("Lars", "Svenson"); c4 =
null;
c3 = c2;
System.out.println (Customer.getCount());
}
```



```
}
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. 5

Answer: D

NEW QUESTION 10

Given the code fragment:

```
Connection con = null;
try {
    // line n1
    if(con != null){
        System.out.print("Connection Established.");
    }

} catch (Exception e) {
    System.out.print(e);
}
```

Assume that dbURL, userName, and password are valid.

Which code fragment can be inserted at line n1 to enable the code to print Connection Established?

- A. Properties prop = new Properties(); prop.put ("user", userName); prop.put ("password", password); con = DriverManager.getConnection (dbURL, prop);
- B. con = DriverManager.getConnection (userName, password, dbURL);
- C. Properties prop = new Properties(); prop.put ("userid", userName); prop.put ("password", password); prop.put("url", dbURL); con = DriverManager.getConnection (prop);
- D. con = DriverManager.getConnection (dbURL); con.setClientInfo ("user", userName); con.setClientInfo ("password", password);

Answer: A

NEW QUESTION 13

You want to create a singleton class by using the Singleton design pattern. Which two statements enforce the singleton nature of the design? (Choose two.)

- A. Make the class static.
- B. Make the constructor private.
- C. Override equals() and hashCode() methods of the java.lang.Object class.
- D. Use a static reference to point to the single instance.
- E. Implement the Serializable interface.

Answer: BD

NEW QUESTION 14

Given the code fragments:

```
4. void doStuff() throws ArithmeticException, NumberFormatException, Exception
{
5. if (Math.random() > -1 throw new Exception ("Try again"); 6. }
and
24. try {
25. doStuff ( );
26. } catch (ArithmeticException | NumberFormatException | Exception e) {
27. System.out.println (e.getMessage()); }
28. catch (Exception e) {
29. System.out.println (e.getMessage()); }
30. }
```

Which modification enables the code to print Try again?

- A. Comment the lines 28, 29 and 30.
- B. Replace line 26 with: } catch (Exception | ArithmeticException | NumberFormatException e) {
- C. Replace line 26 with: } catch (ArithmeticException | NumberFormatException e) {
- D. Replace line 27 with: throw e;

Answer: C

NEW QUESTION 16

Given the code fragments:

```
public class Book implements Comparator<Book> { String name;
double price; public Book () {}
public Book(String name, double price) { this.name = name;
this.price = price;
```

```
}  
public int compare(Book b1, Book b2) { return b1.name.compareTo(b2.name);  
}  
public String toString() { return name + ":" + price;  
}  
}  
and  
List<Book>books = Arrays.asList (new Book ("Beginning with Java", 2), new book ("A  
Guide to Java Tour", 3));  
Collections.sort(books, new Book()); System.out.print(books);  
What is the result?
```

- A. [A Guide to Java Tour:3.0, Beginning with Java:2.0]
- B. [Beginning with Java:2, A Guide to Java Tour:3]
- C. A compilation error occurs because the Book class does not override the abstract method compareTo().
- D. An Exception is thrown at run time.

Answer: A

NEW QUESTION 21

Given:

```
class Counter extends Thread {  
    int i = 10;  
    public synchronized void display(Counter obj) {  
        try {  
            Thread.sleep(5);  
            obj.increment(this);  
            System.out.println(i);  
        } catch (InterruptedException ex) { }  
    }  
    public synchronized void increment (Counter obj) {  
        i++;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        final Counter obj1 = new Counter();  
        final Counter obj2 = new Counter();  
        new Thread(new Runnable() {  
            public void run() {obj1.display(obj2);  
            }  
        }).start();  
        new Thread(new Runnable() {  
            public void run() { obj2.display(obj1); }  
        }).start();  
    }  
}
```

From what threading problem does the program suffer?

- A. race condition
- B. deadlock
- C. starvation
- D. livelock

Answer: B

NEW QUESTION 26

Given that data.txt and alldata.txt are accessible, and the code fragment:

```
public void writeFiles() throws IOException {
    BufferedReader br = new BufferedReader(new FileReader("data.txt"));
    BufferedWriter bw = new BufferedWriter(new FileWriter("alldata.txt"));
    String line = null;
    while ((line = br.readLine()) != null) {
        bw.append(line + "\n");
    }
    // line n1
}
```

What is required at line n1 to enable the code to overwrite alldata.txt with data.txt?

- A. br.close();
- B. bw.writeIn();
- C. br.flush();
- D. bw.flush();

Answer: D

NEW QUESTION 30

Given:

```
public interface LengthValidator {
    public boolean checkLength(String str);
}
```

and

```
public class Txt {
    public static void main(String[] args) {
        boolean res = new LengthValidator() {
            public boolean checkLength(String str) {
                return str.length() > 5 && str.length() < 10;
            }
        }.checkLength("Hello");
    }
}
```

Which interface from the java.util.function package should you use to refactor the class Txt?

- A. Consumer
- B. Predicate
- C. Supplier
- D. Function

Answer: C

NEW QUESTION 35

Given:

```
public class Canvas implements Drawable { public void draw () { }
}
public abstract class Board extends Canvas { }
public class Paper extends Canvas { protected void draw (int color) { }
}
public class Frame extends Canvas implements Drawable { public void resize () { }
}
public interface Drawable { public abstract void draw ();
}
```

Which statement is true?

- A. Board does not compile.
- B. Paper does not compile.
- C. Frame does not compile.
- D. Drawable does not compile.
- E. All classes compile successfully.

Answer: E

NEW QUESTION 39

Given the code fragment:

```
LocalDate valentinesDay =LocalDate.of(2015, Month.FEBRUARY, 14);
LocalDate nextYear = valentinesDay.plusYears(1);
nextYear.plusDays(15); //line n1
```


System.out.println(nextYear); What is the result?

- A. 2016-02-14
- B. A DateTimeException is throw
- C. 2016-02-29
- D. A compilation error occurs at line n1.

Answer: A

NEW QUESTION 40

Given the code fragment:

```
List<Integer> nums = Arrays.asList (10, 20, 8); System.out.println (  
//line n1  
);
```

Which code fragment must be inserted at line n1 to enable the code to print the maximum number in the nums list?

- A. nums.stream().max(Comparator.comparing(a -> a)).get()
- B. nums.stream().max(Integer : : max).get()
- C. nums.stream().max()
- D. nums.stream().map(a -> a).max()

Answer: A

NEW QUESTION 41

Given the definition of the Country class: public class country {

```
public enum Continent {ASIA, EUROPE} String name;
```

```
Continent region;
```

```
public Country (String na, Continent reg) { name = na, region = reg;
```

```
}
```

```
public String getName () {return name;} public Continent getRegion () {return region;}
```

```
}
```

and the code fragment:

```
List<Country> couList = Arrays.asList (
```

```
new Country ("Japan", Country.Continent.ASIA), new Country ("Italy", Country.Continent.EUROPE),
```

```
new Country ("Germany", Country.Continent.EUROPE)); Map<Country.Continent, List<String>> regionNames = couList.stream ()
```

```
.collect(Collectors.groupingBy (Country ::getRegion, Collectors.mapping(Country::getName, Collectors.toList()))); System.out.println(regionNames);
```

- A. {EUROPE = [Italy, Germany], ASIA = [Japan]}
- B. {ASIA = [Japan], EUROPE = [Italy, Germany]}
- C. {EUROPE = [Germany, Italy], ASIA = [Japan]}
- D. {EUROPE = [Germany], EUROPE = [Italy], ASIA = [Japan]}

Answer: B

NEW QUESTION 44

Given:

```
class RateOfInterest {
```

```
public static void main (String[] args) { int rateOfInterest = 0;
```

```
String accountType = "LOAN"; switch (accountType) {
```

```
case "RD"; rateOfInterest = 5; break;
```

```
case "FD"; rateOfInterest = 10; break;
```

```
default:
```

```
assert false: "No interest for this account"; //line n1
```

```
}
```

```
System.out.println ("Rate of interest:" + rateOfInterest);
```

```
}
```

```
}
```

and the command:

```
java -ea RateOfInterest What is the result?
```

- A. Rate of interest: 0
- B. An AssertionError is thrown.
- C. No interest for this account
- D. A compilation error occurs at line n1.

Answer: B

NEW QUESTION 47

Given the code fragment:

```
public static void main(String[] args) {  
    Console console = System.console();  
    char[] pass = console.readPassword("Enter password:"); // line n1  
    String password = new String(pass); // line n2  
}
```

What is the result?

- A. A compilation error occurs at line n1.
- B. A compilation error occurs at line n2.
- C. The code reads the password without echoing characters on the console.
- D. A compilation error occurs because the IOException isn't declared to be thrown or caught?

Answer: D

NEW QUESTION 49

Given the code fragment:

```
// Login time:2015-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(1000);

// Logout time:2015-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES);    // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (logoutTime.isAfter(loginTime))
    System.out.println("Logged out at:"+logoutTime);
else
    System.out.println("Can't logout");
```

What is the result?

- A. A compilation error occurs at line n1.
- B. Logged out at: 2015-01-12T21:58:19.880Z
- C. Can't logout
- D. Logged out at: 2015-01-12T21:58:00Z

Answer: D

NEW QUESTION 52

Given:

```
class Vehicle { int vno;
String name;
public Vehicle (int vno, String name) { this.vno = vno;;
this.name = name;
}
public String toString () { return vno + ":" + name;
}
}
```

and this code fragment:

```
Set<Vehicle> vehicles = new TreeSet <> (); vehicles.add(new Vehicle (10123, "Ford")); vehicles.add(new Vehicle (10124, "BMW")); System.out.println(vehicles);
```

What is the result?

- A. 10123 Ford10124 BMW
- B. 10124 BMW10123 Ford
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

Answer: D

NEW QUESTION 54

Given the code fragment:

```
9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
10. String query = "SELECT id FROM Employee";
11. try (Statement stmt = conn.createStatement()) {
12. ResultSet rs = stmt.executeQuery(query);
13. stmt.executeQuery("SELECT id FROM Customer");
14. while (rs.next()) {
15. //process the results
16. System.out.println("Employee ID: "+ rs.getInt("id"));
17. }
18. } catch (Exception e) {
19. System.out.println ("Error");
20. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. compilation fails on line 13.

Answer: C

NEW QUESTION 58

Which statement is true about java.util.stream.Stream?

- A. A stream cannot be consumed more than once.
- B. The execution mode of streams can be changed during processing.
- C. Streams are intended to modify the source data.
- D. A parallel stream is always faster than an equivalent sequential stream.

Answer: B

NEW QUESTION 60

Given that version.txt is accessible and contains: 1234567890

and given the code fragment:

```
try (FileInputStream fis = new FileInputStream("version.txt");
    InputStreamReader isr = new InputStreamReader(fis);
    BufferedReader br = new BufferedReader(isr);) {
    if (br.markSupported()) {
        System.out.print((char) br.read());
        br.mark(2);
        System.out.print((char) br.read());
        br.reset();
        System.out.print((char) br.read());
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

What is the result?

- A. 121
- B. 122
- C. 135
- D. The program prints nothing.

Answer: B

NEW QUESTION 62

Given the code fragment:

```
class CallerThread implements Callable<String> { String str;
public CallerThread(String s) {this.str=s;} public String call() throws Exception { return str.concat("Call");
}
}
and
public static void main (String[] args) throws InterruptedException, ExecutionException
{
    ExecutorService es = Executors.newFixedThreadPool(4); //line n1 Future f1 = es.submit (newCallerThread("Call"));
    String str = f1.get().toString(); System.out.println(str);
}
```

Which statement is true?

- A. The program prints Call Call and terminates.
- B. The program prints Call Call and does not terminate.
- C. A compilation error occurs at line n1.
- D. An ExecutionException is thrown at run time.

Answer: B

NEW QUESTION 67

Given the definition of the Employee class:


```
class Employee {
    String dept, name;
    public Employee(String d, String n) {
        dept = d;
        name = n;
    }
    public String toString() {
        return getDept() + ":" + getName();
    }
    public String getDept() { return dept; }
    public String getName() { return name; }
}
```

and this code fragment:

```
List<Employee> emps = Arrays.asList(new Employee("sales", "Ada"),
    new Employee("sales", "Bob"),
    new Employee("hr", "Bob"),
    new Employee("hr", "Eva"));
Stream<Employee> s = emps.stream()
    .sorted(Comparator.comparing((Employee e) -> e.getDept())
        .thenComparing((Employee e) -> e.getName()));
List<Employee> eSorted = s.collect(Collectors.toList());
System.out.println(eSorted);
```

What is the result?

- A. [sales:Ada, hr:Bob, sales:Bob, hr:Eva]
- B. [Ada:sales, Bob:sales, Bob:hr, Eva:hr]
- C. [hr:Eva, hr:Bob, sales:Bob, sales:Ada]
- D. [hr:Bob, hr:Eva, sales:Ada, sales:Bob]

Answer: A

NEW QUESTION 68

Given:

```
class Product {
    String pname;
    public Product(String pname) {
        this.pname = pname;
    }
}
```

and the code fragment:

```
Product p1 = new Product("PowerCharger");
Product p2 = p1;
System.out.println(p1.equals(p2));
Product p3 = new Product("PowerCharger");
System.out.println(p1.equals(p3));
```

What is the result?

- A. true true
- B. false true
- C. false false
- D. true false

Answer: B

NEW QUESTION 71

Given the code fragments:

```
class Employee { Optional<Address> address;
Employee (Optional<Address> address) { this.address = address;
}
public Optional<Address> getAddress() { return address; }
}
class Address {
String city = "New York";
public String getCity { return city; } public String toString() {
return city;
}
}
and
Address address = null;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not available";
What is the result?
```

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

Answer: B

NEW QUESTION 73

Given:

```
class ImageScanner implements AutoCloseable { public void close () throws Exception { System.out.print ("Scanner closed.");
}
public void scanImage () throws Exception { System.out.print ("Scan.");
throw new Exception("Unable to scan.");
}
}
class ImagePrinter implements AutoCloseable { public void close () throws Exception { System.out.print ("Printer closed.");
}
public void printImage () {System.out.print("Print."); }
}
and this code fragment:
try (ImageScanner ir = new ImageScanner(); ImagePrinter iw = new ImagePrinter()) { ir.scanImage();
iw.printImage();
} catch (Exception e) { System.out.print(e.getMessage());
}
What is the result?
```

- A. Scan.Printer close
- B. Scanner close
- C. Unable to scan.
- D. Scan.Scanner close
- E. Unable to scan.
- F. Sca
- G. Unable to scan.
- H. Sca
- I. Unable to sca
- J. Printer closed.

Answer: A

NEW QUESTION 78

Given the code fragments:

```
class R implements Runnable {
    public void run() { System.out.println("Run..."); }
}

class C implements Callable<String> {
    public String call() throws Exception { return "Call..."; }
}
```

and


```
ExecutorService es = Executors.newSingleThreadExecutor();
es.execute(new R()); // line n1
Future<String> f1 = es.submit(new C()); // line n2
System.out.println(f1.get());
es.shutdown();
```

What is the result?

- A. The program prints Run... and throws an exception.
- B. A compilation error occurs at line n1.
- C. Run...Call...
- D. A compilation error occurs at line n2.

Answer: B

NEW QUESTION 82

Given the code fragment:

```
List<Integer> li = Arrays.asList(10, 20, 30);
Function<Integer, Integer> fn = f1 -> f1 + f1;
Consumer<Integer> conVal = s -> System.out.print("Val:" + s + " ");
li.stream().map(fn).forEach(conVal);
```

What is the result?

- A. Val:20 Val:40 Val:60
- B. Val:10 Val:20 Val:30
- C. A compilation error occurs.
- D. Val: Val: Val:

Answer: B

NEW QUESTION 85

Given:

```
class Person {
    String name;
    int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName(){ return name; }
    public int getAge(){ return age; }
}
```

and the code fragment:

```
List<Person> sts = Arrays.asList(
    new Person("Jack", 30),
    new Person("Mike Hill", 21),
    new Person("Thomas Hill", 24));
Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 25); // line n1
long count = resList.filter(s -> s.getName().contains("Hill")).count();
System.out.print(count);
```

What is the result?

- A. A compilation error occurs at line n1.
- B. An Exception is thrown at run time.
- C. 2

Answer: B

NEW QUESTION 88

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your 1Z0-809 Exam with Our Prep Materials Via below:

<https://www.certleader.com/1Z0-809-dumps.html>